



Pennsylvania State University

Abington Campus

Abington, PA, 19001

Final Project

Customer Segmentation

Besjana Kubick

Cameron Bussom

Mohamed Chikani

CMPSC 446, Section 001: Data Mining

Dr. MD Hossain

December 9, 2023

Table of contents

I. Introduction	3
Problem Statement MC add specific Aims/Overview	3
Objective	3
Motivation	3
Related Work	3
II. Data	4
Data Format:	4
III. Methodology	5
Technical Approach	5
Setting up the data	6
Preprocessing and cleaning	6
Data visualization	7
Procedures and Features	8
Features	9
IV. Experiments	10
Parameter Tuning	10
Evaluation Metrics	11
Analysis of Results	16
V. Conclusion	16
VI. Instructions	16

I. Introduction

Problem Statement

Despite having a wealth of customer data, companies often lack the insights needed to effectively segment customers, resulting in generalized marketing strategies that fail to address specific customer needs. This project aims to address those inefficiencies in a one-size-fits-all approach to customer relationship management. Customizing interactions based on segmented customer groups can lead to increased satisfaction and profitability.

Objective

To develop and implement a sophisticated machine learning model, specifically utilizing the K-means clustering algorithm, to segment the customer base into distinct groups. These groups will be based on quantifiable criteria such as purchase frequency, average transaction value, product category preferences. This segmentation aims to enable the development of targeted marketing strategies, personalized product recommendations, and improved customer engagement, leading to an anticipated increase in customer retention rates.

Motivation

The primary motivation for undertaking this project is educational, as part of an introduction to data mining course. The project serves as a practical application to deepen understanding and skills in data mining techniques, particularly in the realm of unsupervised learning. By focusing on customer segmentation using machine learning, the project offers hands-on experience in managing real-world data, exploring the nuances of clustering algorithms like K-means, and understanding their practical applications in business contexts. This endeavor will not only solidify theoretical knowledge acquired in the course but also provide valuable insights into the versatility and impact of data mining in solving complex marketing challenges and driving data-driven decision-making in commercial settings.

Related Work

Review of existing literature and methodologies on customer segmentation, with an emphasis on machine learning approaches. This project seeks to build upon the foundation of existing research, applying it within a new dataset to gain hands-on experience in data mining procedures.

II. Data

Data Format:

The data will primarily consist of customer transaction records and interaction logs in CSV or JSON format.

1. **Product Title:** Title of the purchased item
2. **Product Type:** Type or category of the purchased item
3. **Variant Title:** Title of the variant (if applicable)
4. **Variant SKU:** Stock Keeping Unit for the variant
5. **Variant ID:** Unique identifier for the variant
6. **Customer ID:** Unique identifier for each customer
7. **Order ID:** Unique identifier for each order
8. **Day:** Date of the purchase
9. **Net Quantity:** Net quantity of the purchased item
10. **Gross Sales:** Gross sales amount for the item
11. **Discounts:** Discounts applied to the purchase
12. **Returns:** Amount refunded for returned items
13. **Net Sales:** Net sales amount after discounts and returns
14. **Taxes:** Taxes applied to the purchase
15. **Total Sales:** Total sales amount including taxes
16. **Returned Item Quantity:** Quantity of items returned by the customer
17. **Ordered Item Quantity:** Total quantity of items ordered by the customer

Dataset Structure Example

product_title		product_type	variant_title		variant_sku		variant_id	customer_id	order_id	day
0	DPR	DPR	100		AD-982-708-895-F-6C894FB		52039657	1312378	83290718932496	04/12/2018
1	RJF	Product P	28 / A / MTM		83-490-E49-8C8-8-3B100BC		56914686	3715657	36253792848113	01/04/2019
2	CLH	Product B	32 / B / FtO		68-ECA-BC7-3B2-A-E73DE1B		24064862	9533448	73094559597229	05/11/2018
3	NMA	Product F	40 / B / FtO	6C-1F1-226-1B3-2-3542B41		43823868	4121004	53616575668264		19/02/2019
4	NMA	Product F	40 / B / FtO	6C-1F1-226-1B3-2-3542B41		43823868	4121004	29263220319421		19/02/2019
<div>< <div></div></div>										
net_quantity	gross_sales	discounts	returns	net_sales	taxes	total_sales	returned_item_quantity	ordered_item_quantity		
2	200.0	-200.00	0.00	0.0	0.0	0.0		0		
2	190.0	-190.00	0.00	0.0	0.0	0.0		0		
0	164.8	-156.56	-8.24	0.0	0.0	0.0		-2		
1	119.0	-119.00	0.00	0.0	0.0	0.0		0		
1	119.0	-119.00	0.00	0.0	0.0	0.0		0		

The dataset consists of 70052 entries of customer orders.

```
RangeIndex: 70052 entries, 0 to 70051
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   product_title          70052 non-null  object
1   product_type           70052 non-null  object
2   variant_title          70052 non-null  object
3   variant_sku            70052 non-null  object
4   variant_id             70052 non-null  int64
5   customer_id            70052 non-null  int64
6   order_id               70052 non-null  int64
7   day                    70052 non-null  object
8   net_quantity           70052 non-null  int64
9   gross_sales            70052 non-null  float64
10  discounts              70052 non-null  float64
11  returns                70052 non-null  float64
12  net_sales              70052 non-null  float64
13  taxes                  70052 non-null  float64
14  total_sales            70052 non-null  float64
15  returned_item_quantity 70052 non-null  int64
16  ordered_item_quantity  70052 non-null  int64
dtypes: float64(6), int64(6), object(5)
memory usage: 9.1+ MB
```

III. Methodology

Technical Approach

Language and Libraries: The project will primarily utilize Python, known for its robust data science capabilities.


Key libraries include:

- *Pandas* for data manipulation and analysis.
- *NumPy* for numerical computations.
- *Scikit-learn* for implementing the K-means clustering algorithm and other machine learning tools.
- *Matplotlib* and *Seaborn* for data visualization.

Data Processing Frameworks: Utilizing Jupyter Notebooks for interactive data exploration and processing.

Setting up the data

	variant_id	customer_id	order_id	net_quantity	gross_sales	discounts	returns
count	7.005200e+04	7.005200e+04	7.005200e+04	70052.000000	70052.000000	70052.000000	70052.000000
mean	2.442320e+11	6.013091e+11	5.506075e+13	0.701179	61.776302	-4.949904	-10.246051
std	4.255079e+12	6.223201e+12	2.587640e+13	0.739497	31.800689	7.769972	25.154677
min	1.001447e+07	1.000661e+06	1.000657e+13	-3.000000	0.000000	-200.000000	-237.500000
25%	2.692223e+07	3.295695e+06	3.270317e+13	1.000000	51.670000	-8.340000	0.000000
50%	4.494514e+07	5.566107e+06	5.522207e+13	1.000000	74.170000	0.000000	0.000000
75%	7.743106e+07	7.815352e+06	7.736876e+13	1.000000	79.170000	0.000000	0.000000
max	8.422212e+13	9.977409e+13	9.999554e+13	6.000000	445.000000	0.000000	0.000000



The dataset reflects a diverse range of customer transactions, with notable variations in key features. On average, customers tend to order a small quantity of items, with a mean net quantity of approximately 0.70. The gross sales values exhibit moderate variability, suggesting a mix of high and low-value transactions. Notably, returns contribute to a reduction in net sales, and their presence suggests potential challenges in customer satisfaction or product suitability.

A quick statistical overview will show us that there are a significant number of rows whose `ordered_item_quantity` is 0 and `net_quantity` is less than 0, which means they are not ordered/sold at all, as seen above. These rows (which amount to 10715) will be excluded from the orders dataset for the project.

Preprocessing and cleaning

- The initial exploration of the dataset in the section above revealed that several steps were needed to preprocess and clean the data for meaningful analysis. A significant number of rows whose `ordered_item_quantity` is 0 and `net_quantity` is less than 0, which means they are not ordered/sold at all, as seen above. These rows (which amount to 10715) will be excluded from the orders dataset for the project. This step ensured that only relevant transactions were considered.
- The dataset has each sale separately. For our purposes we needed to see how many products each customer ordered. The `'products_ordered'` feature was calculated, using a function that was devised to aggregate the data by counting the number of ordered items for each customer, considering the `'product_type'`

column. This function identified customers who ordered multiple products and resulted in a new feature indicating the count of products ordered.

- The 'average_return_rate' feature was engineered by calculating the ratio of returned item quantity to ordered item quantity. This ratio was averaged for all orders of a customer, providing insights into the likelihood of returns across various transactions.
- Total spending, a critical feature, was computed by aggregating the sum of 'total_sales' per customer. This metric considered the amount after taxes and returns, providing a comprehensive view of a customer's overall spending.
- To address skewed distributions and enhance model performance, a log transformation was applied to the features. The logarithmic transformation was particularly useful for 'products_ordered' and 'total_spending' columns, mitigating the impact of extreme values.
- Rows with negative 'net_quantity,' indicating returns without corresponding orders, were excluded. This step refined the dataset, focusing on genuine customer transactions.

The final cleaned dataset, now containing meaningful features like 'products_ordered,' 'average_return_rate,' and 'total_spending,' is prepared for subsequent analysis and model application. The log transformation ensures that the features are better suited for clustering algorithms, providing a normalized and improved representation of customer behavior.

Data visualization

- **Histograms for Feature Distribution:**
 - Purpose: To understand the distribution of key features such as products_ordered, average_return_rate, and total_spending.
 - Implementation: Using Plotly's go.Histogram to create interactive histograms.
 - Insights: These histograms help identify patterns and outliers in the data, and can suggest transformations needed for clustering.
- **3D Scatter Plots for Cluster Visualization:**
 - Purpose: To visualize the clusters formed by the K-means algorithm in a three-dimensional space.

- Implementation: Plotly's `scatter_3d` is used to plot the clusters based on transformed features like `log_products_ordered`, `log_average_return_rate`, and `log_total_spending`.
- Insights: This visualization provides a clear picture of how the clusters are distributed and separated in the multidimensional feature space.
- **Elbow Method Visualization:**
 - Purpose: To find the optimal number of clusters for K-means clustering.
 - Implementation: A line plot showing the relationship between the number of clusters
 - Insights: The 'elbow' point on the graph helps in determining the number of clusters at which the within-cluster sum of squares (WCSS) starts to decrease at a slower rate.
- **Bar Chart for Cluster Cardinality:**
 - Purpose: To visualize the number of data points in each cluster.
 - Implementation: Using a bar chart to show the magnitude of each cluster.
 - Insights: This helps in understanding the relative sizes of the clusters, which is important for assessing the balance and practical significance of each segment.
- **Dendrogram for Hierarchical Clustering:**
 - Purpose: To visualize the tree-like structure of data points created by hierarchical clustering.
 - Implementation: Utilizing dendrogram from Scipy's library.
 - Insights: A dendrogram helps in deciding the number of clusters by showing the point at which clusters merge, indicating the similarity between data points.

Procedures and Features

- **Data Preprocessing:**
 - Loading Data: Importing customer orders data into a pandas DataFrame.
 - Cleaning and Inspection: Removing entries with negative net quantities and examining basic dataset structures.
 - Descriptive Statistics: Generating statistical summaries for non-object columns to understand data distribution and scale.
- **Feature Engineering:**
 - Aggregation by Ordered Quantity: Grouping data by customer and product type to count ordered item quantities.
 - Average Return Rate: Calculating the ratio of returned to ordered items for each order, then averaging this ratio for each customer.

- Total Spending Calculation: Summing total sales per customer to determine their total spending.
- **Data Transformation:**
 - Logarithmic Transformation: Applying `np.log1p` to key features (`products_ordered`, `average_return_rate`, `total_spending`) to normalize data and reduce skewness.
- **K-Means Clustering:**
 - Model Initialization: Creating an initial K-means model with standard parameters.
 - Model Fitting: Fitting the model to the transformed features.
 - Inertia Calculation: Computing the within-cluster sum-of-squares (inertia) to evaluate compactness.
- **Hyperparameter Tuning (Elbow Method):**
 - Inertia Analysis: Running K-means with different values of K and recording the inertia to find the optimal number of clusters.
 - Visualization: Plotting the inertia values against K values to identify the 'elbow point.'
- **Updated K-Means Model:**
 - Model Creation: Building a new K-means model with the determined optimal number of clusters.
 - Cluster Prediction: Predicting cluster labels for the dataset.
- **Hierarchical Clustering:**
 - Linkage Matrix Creation: Using the ward method to create a hierarchical clustering.
 - Dendrogram Visualization: Plotting a dendrogram to visualize the hierarchical clustering process.
 - Cluster Labeling: Applying Agglomerative Clustering to assign cluster labels to the data.

Features

- **Original Features:**
 - Products Ordered (`products_ordered`): Count of different products ordered by each customer.

- Total Spending (total_spending): Total amount spent by each customer.
- **Total Spending:**
 - This is a direct measure of a customer's financial engagement with the business. By analyzing total spending, we can gain insights into customer value and purchasing power.
- **Average Total Spending:**
 - Calculating the average spending over a given period can help normalize the data and provide a clearer view of spending habits.
- **Average Return Rate:**
 - This metric gives an insight into the satisfaction and preferences of customers. A high return rate might indicate dissatisfaction or issues with the product mix.
- **Algorithm Overview:**
 - K-means is a centroid-based clustering algorithm that partitions the data into K clusters. Each data point is assigned to the cluster with the nearest mean, serving as a prototype of the cluster.
- **Feature Selection for K-Means:**
 - The features chosen for clustering, such as average total spending and average return rate, should be reflective of customer behaviors and characteristics that are relevant for segmentation.

IV. Experiments

Parameter Tuning

Parameter tuning in the context of K-means clustering is pivotal for the effective segmentation of the customer base. The process involves fine-tuning the algorithm's parameters to ensure that the model accurately captures the inherent groupings in the data. For this project, the focus is on the customers' purchase history, particularly variables like total spending, from which we can derive insightful features such as average total spending and average return rate.

Evaluation Metrics

Application in Customer Segmentation

- **Selecting Appropriate Metrics:**
 - Given the nature of customer data and the business objectives, it might be beneficial to consider multiple metrics for a comprehensive evaluation.
- **Comparison with Business Understanding:**
 - The results from these metrics should be aligned with the business understanding of the customer base. For instance, segments should make intuitive sense and reflect recognizable patterns in customer behavior.
- **Balancing Statistical and Business Objectives:**
 - While low WCSS and high silhouette scores are desirable, they should be balanced with the practicality of the number of segments and their interpretability in a business context.

Post-Evaluation Steps

- **Cluster Profiling:**
 - Once the clusters are validated using these metrics, detailed profiling of each cluster is essential to understand the common characteristics of customers within each segment.

Model Implementation

- Implementing K-means clustering algorithm using Scikit-learn.
- Determining the optimal number of clusters (k) using the Elbow Method and
- Iteratively training the model on the preprocessed dataset.

Model Evaluation

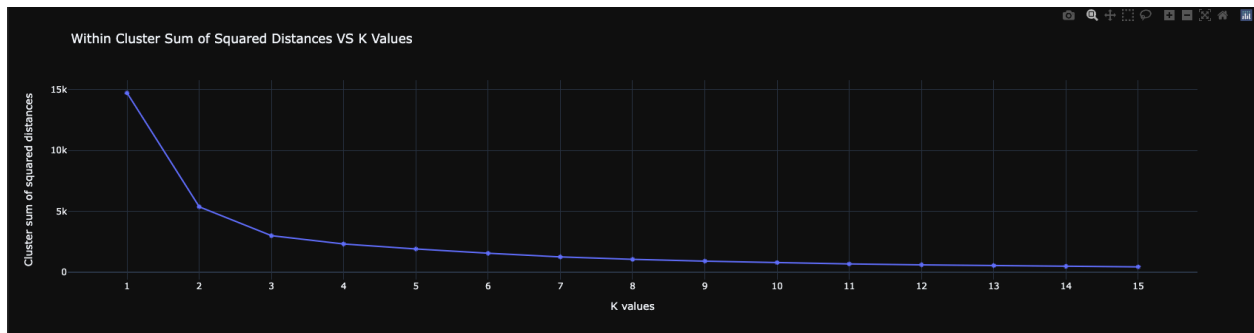
- Assessing the performance of the clustering model using metrics like Within-Cluster Sum of Squares (WCSS). In the context of customer segmentation, minimizing WCSS means that the customers in each cluster are as close as possible to their respective cluster centroids. A lower WCSS indicates tighter and more defined clusters. However, it's important to balance it with the number of clusters to avoid overfitting (where increasing the number of clusters indefinitely reduces WCSS)
- Using Scikit-learn to visualize clusters

Post-Modeling Analysis

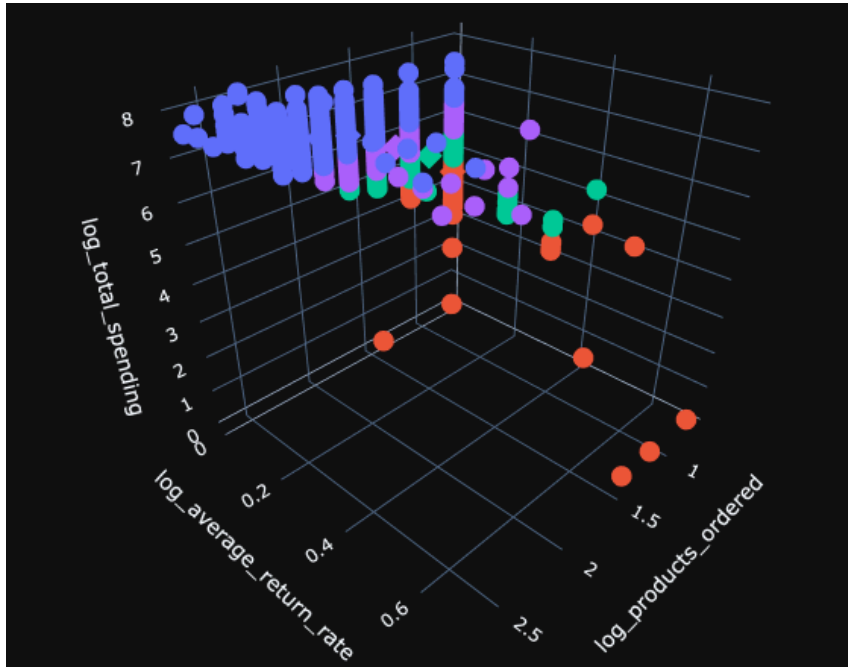
- Profiling each cluster to understand the defining characteristics of each customer segment.
- Performing statistical analysis to validate the significance of the differences between clusters.

Experiments Comparison

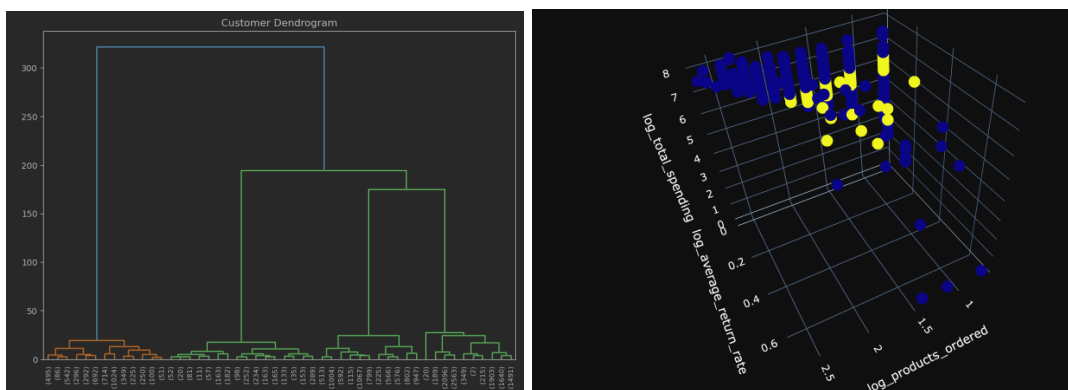
In summary we did two experiments to do a customer segmentation. We used K-means clustering and Hierarchical Clustering. This was necessary to make sure our results were robust and can be used in a business setting. For the K-means clustering we used Hyperparameter Tuning, the Elbow method. After applying log transformation on the new engineered features, we proceeded with the segmentation by feeding the new dataframe into the K-mean Clustering. K-means clustering works by assigning each data point to the closest centroid based on euclidean distance. This is where the question arises: “How well our dataset was clustered by K-Means?”. To answer this we used the Elbow Method as our Hyperparameter tuning. The elbow method revealed that the optimal number of clusters is 4 since the reduction in variance slowed down significantly after K=4.



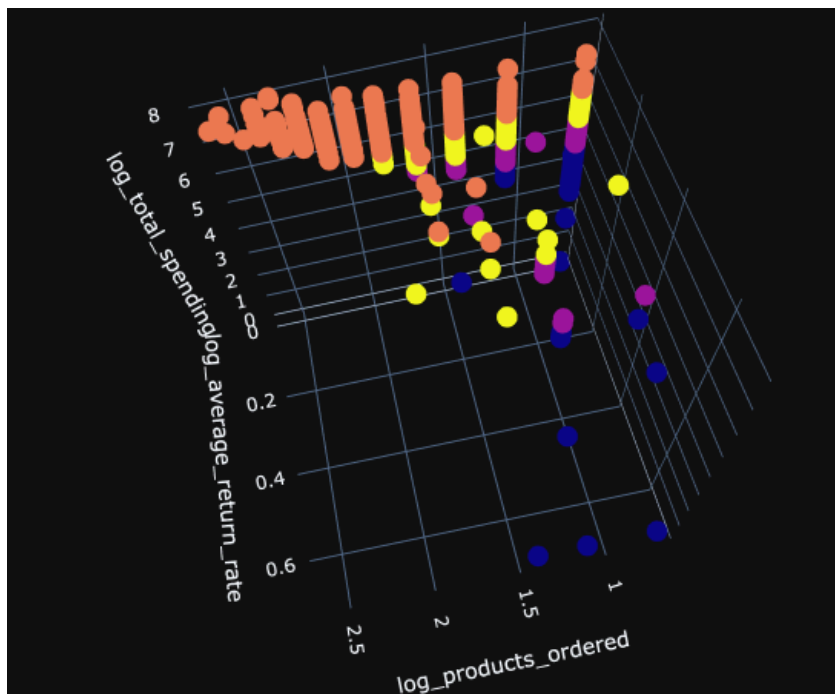
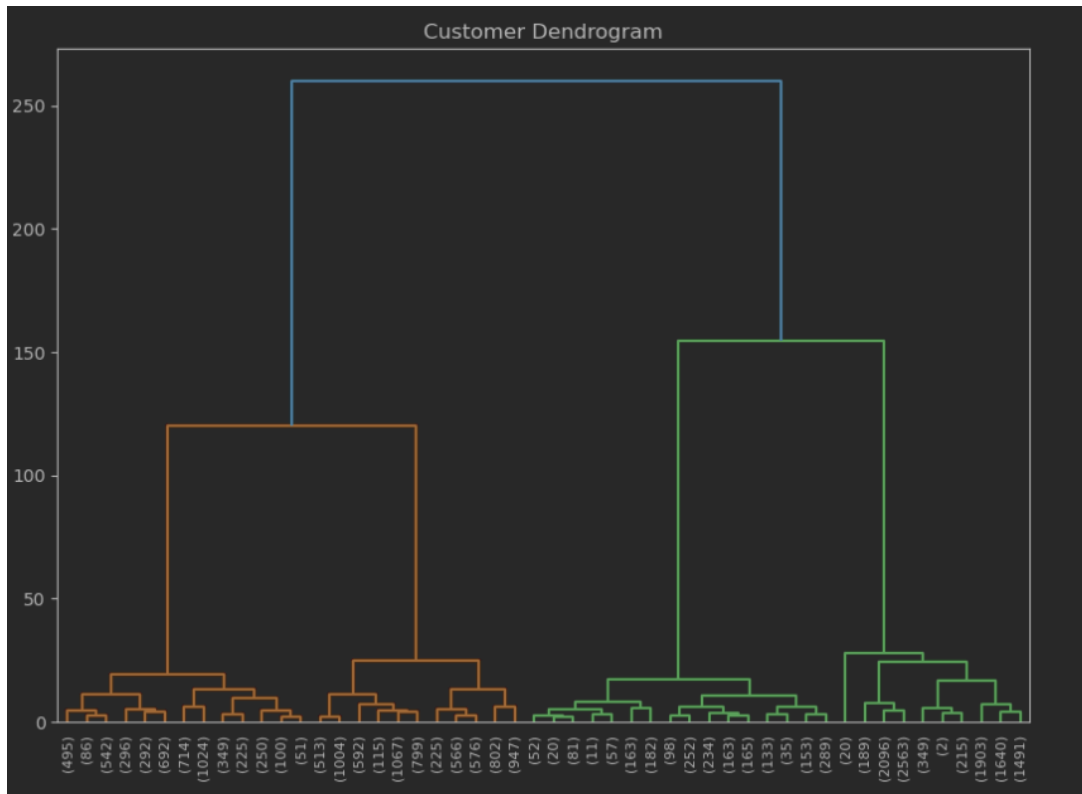
Once we updated the optimal number of clusters we applied a 3D Scatter plot to the K-Mean clustering to visualize the created clusters. Data points are represented as spheres and one can see detailed information by hovering over. This will make it easier to do a post evaluation analysis.



As mentioned above, to be able to assess consistency and validate robustness of the defined clusters, we also applied Hierarchical Clustering to our data. The Agglomerative clustering builds a tree of clusters without first specifying the number of clusters. After applying the Agglomerative Clustering without specifying the number of clusters we then performed data visualization using 3D scatter plots which showed two big data clusters. As shown below:



This implied that we would only have two big clusters. Considering how we had a large amount of customer data, and segmenting them in only two big clusters would not be specific and actionable enough, we decided to experiment and change the number of clusters. This meant we had to cut off the tree structure at a different point. The obvious choice as seen in the picture above, was to set the number of clusters to 4. Higher than 4 would result in too many clusters and less than 4 would not provide enough relevant information.

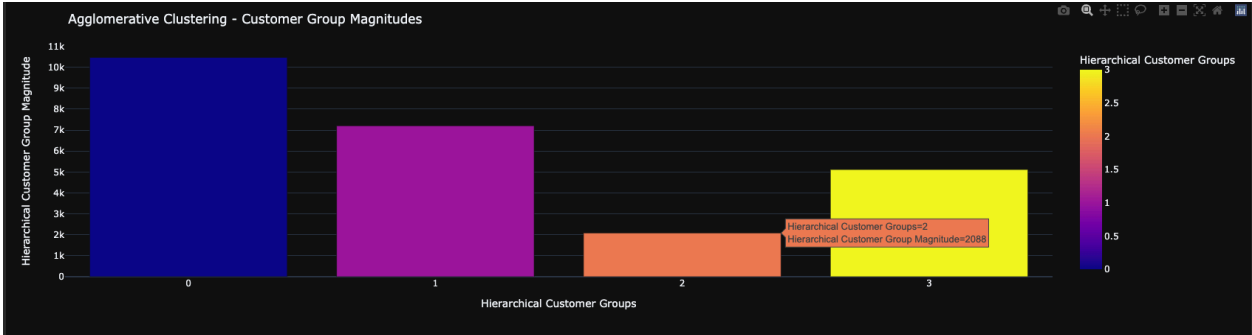


Upon finishing the clustering we checked cluster magnitudes to gain an overview on the insights that would be usable in a business setting. You can see the results below.

K-Means Customer Groups		K-Means Customer Group Magnitude
0	1	10468
1	2	7206
2	3	5116
3	0	2088



Hierarchical Customer Groups		Hierarchical Customer Group Magnitude
0	0	10468
1	1	7206
2	3	5116
3	2	2088



Analysis of Results

As we can see from the Customer Group Magnitude above, this analysis created four customer segmentations. From analyzing the 3D plot and the customer group magnitude histogram for the K-mean we can see that cluster 0 (light blue) is the most desirable group of customers and makes up for only 8.376% of the entire customer base of the business. This group spends the most, has the highest amount of products ordered and the lowest average return rate. This customer segment is represented as the Orange group or cluster 2 in the Agglomerative Clustering histogram. Cluster 1 (red for K-means and dark blue for Hierarchical Clustering) makes up for 41.99% of the customer base and is the group who spends the least, returns the most and orders the least amount of products. The two clusters build the other half of the customers.

This analysis can be very useful for a business to be able to apply targeted strategies to increase their revenues. For example, any movement of customers from Cluster 0 (for K-means) or Cluster 2 (for Agglomerative Clustering) to the other segments would improve the revenues. Businesses can change their strategies based on how their customers are segmented. By running this analysis every time a new product or marketing strategy is implemented the company can see how this affects their customers and also analyze if their approach is having the desired effect.

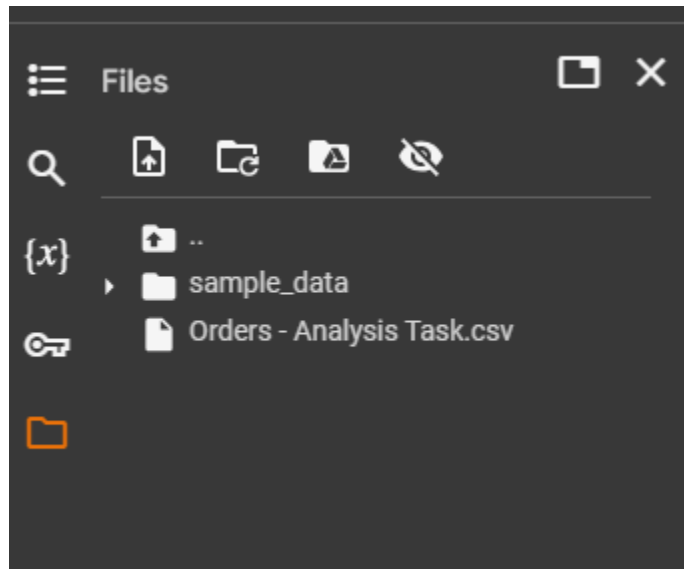
V. Conclusion

Since this was a project for a data mining course and was being used to deepen our knowledge about data mining this was the perfect project to get an overview of different Data Mining techniques and how to analyze the findings. Customer segmentation analysis was used to identify distinct customer segments based on their purchasing behaviors. We got insight into how different customer groups contribute to sales, returns and overall spending. We identified dominant segments by using two different methods that solidified our findings. The results can be translated into targeted and actionable strategies for marketing and sales, or operational improvements potentially leading to more personalized customer engagement and improved resource allocation

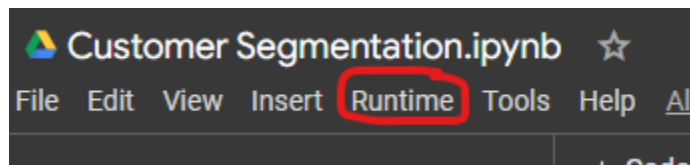
VI. Instructions

- A. To run the Google colab of our code, go to the following link: [Customer Segmentation.ipynb](#). The link gives you access to a shared folder. Once in the folder click on the file and choose open with Google Colab.
- B. When opening the link you need to add Order - Analysis Task.csv file to the file directory, this can be done by clicking on the file tab Colab and right clicking,

then selecting upload. Make sure you have first downloaded the .csv file locally. When in the file explorer select the file and click open. It should look like the image below



- C. Once the file is uploaded you are ready to run the code, select the runtime button at the top of the screen and click Run All, this will run the entire program.



- D. You can also run individual pieces of code by clicking on the play button to the left of the snippets of code.

